LDAPPC

LDAPPC - LDAP Provisioning Connector as of v1.5.0

LDAPPC provisions group and membership information contained in the Groups Registry to an LDAP directory service.

> See also the information on the newer provisioning connector called LDAPPC-NG.
>
> As of Grouper 2.1, see the Provisioning Service Provider (PSP)

Usage

LDAPPC is run using GrouperShell (gsh).

For example, to maintain group and membership provisioning, polling every 60 seconds for changes :

```
bin/gsh.sh -ldappc -groups -memberships -interval 60
```

One or both of `-groups` and `-memberships` must be specified. All other arguments are optional.

Key	Value	Description
no arguments		Display usage.
-groups		Provision groups.
-memberships		Provision memberships.
-subject	*subjectId*	The SubjectId used to establish Grouper API sessions. Defaults to GrouperSystem.
-interval	*interval*	Number of seconds between polling intervals. If omitted, only one provisioning cycle is performed.
-lastModifyTime	*yyyy-MM-dd[_hh:mm:ss]*	Select objects changed since this time.
-configManager	*path to configuration xml*	Path to configuration file. Defaults to classpath resource ldappc.xml.
-properties	*path to properties file*	Path to properties file. Defaults to classpath resource ldappc.properties.
-resolver	*path to directory*	Path to directory containing Shibboleth Attribute Resolver configuration files.
-calc	*file*	Calculate provisioning and write to file.
-dryRun	*file*	Write provisioning changes to file only, do not provision changes.
-logLDIF		While provisioning, log changes in LDIF format.

Release Notes

Version 1.5.0 of LDAPPC includes several new features, many of which were implemented because of requests on the Grouper mailing lists. Thank you for your involvement.

The ability to provision Active Directory has improved significantly. Integration with the Shibboleth Attribute Resolver provides customizable attributes, potentially suitable for Exchange. Integration with vt-ldap 3.2 provides support for paging and groups with a large (>1500) number of members.

An upcoming version of LDAPPC should include SPML 2 support.

- GRP-329 Group attributes calculated by the Shibboleth Attribute Resolver may be provisioned.
- GRP-335 LDAP connectivity is now provided by vt-ldap.
- GRP-333 The <source-subject-identifier source="g:gsa" ...> configuration element is no longer allowed nor necessary.
- GRP-325 Command line option to calculate provisioning and write to file.
- GRP-326 Command line option to write provisioning changes to file, don't execute changes.
- GRP-327 Command line option to log provisioning changes as LDIF.
- GRP-275 Groups are provisioned in two phases to handle member groups.
- GRP-332 When a subject can not be found, LDAPPC can be configured to fail, log, or ignore.
- GRP-330 Multiple objectclasses may be provisioned.
- GRP-328 Configuration macros in ldappc.xml and ldappc.properties are supported.

- GRP-334 The bundling of attribute modifications is configurable.
- GRP-331 Multiple target objects may be provisioned per subject.
- GRP-339 Ignore member groups that are not in the set of groups to be provisioned.
- GRP-346 More advanced query filters are supported when selecting the groups to be provisioned.
- GRP-342 Error when synchronizing a group name containing a forward slash '/'.

Upgrading to LDAPPC 1.5.0

The <ldap> configuration element in ldappc.xml is no longer allowed. LDAP connection parameters are now defined in ldappc.properties. See vt-ldap .

Remove any <source-subject-identifier source="g:gsa" ...> configuration elements.

As of Grouper v1.4.1, LDAPPC is included in the Grouper API. Previously LDAPPC was a separate project.

Configuration

LDAPPC requires two files, `ldappc.xml` and `ldappc.properties`. The full path to these files may be defined at runtime.

By default, macros of the form ${name} in `ldappc.xml` will be replaced by their corresponding values in `ldappc.properties`.

ldappc.properties

LDAP connectivity is provided by vt-ldap and is defined in `ldappc.properties`.

```
# Macros of the form ${name} in your configuration (default ldappc.xml)
# will be replaced with the values of the matching keys of this file.

edu.vt.middleware.ldap.ldapUrl=ldap://127.0.0.1:389
edu.vt.middleware.ldap.base=dc=example,dc=edu
edu.vt.middleware.ldap.authtype=simple
edu.vt.middleware.ldap.serviceUser=cn=Manager
edu.vt.middleware.ldap.serviceCredential=secret
edu.vt.middleware.ldap.tls=true
```

ldappc.xml

- **<ldappc> - Provisioning Configuration**

  ```
  <ldappc>
   <grouper ... />
   <source-subject-identifiers ... />
  </ldappc>
  ```

  The LDAPPC provisioning configuration consists of two elements : Grouper and LDAP subject parameters.
- **<grouper> - Grouper Configuration**

  ```
  <grouper>
   <grouper-queries ... />
   <groups ... />
   <memberships ... />
  </grouper>
  ```

  The Grouper configuration includes the selection of groups to be provisioned, and optional group and membership provisioning.
- **<grouper-queries> - Select Groups to be Provisioned**

  ```
  <grouper-queries>
   <subordinate-stem-queries ... />
   <attribute-matching-queries ... />
   <resolver-matching-queries ... />
  </grouper-queries>
  ```

  The groups to be provisioned may be selected by stem, attribute value, or a union of both. Currently, it is not possible to exclude a group that otherwise matches the selection criteria from being provisioned.
- <subordinate-stem-queries> - Select Groups to be Provisioned by Stem

```
<subordinate-stem-queries>
 <stem-list>
  <stem>uc:faculty:art</stem>
  <stem>uc:faculty:math</stem>
 </stem-list>
</subordinate-stem-queries>
```

All groups subordinate to any of the given stems are selected for provisioning.

- <attribute-matching-queries> - Select Groups to be Provisioned by Attribute

```
<attribute-matching-queries>
 <attribute-list>
  <attribute name="attr1" value="value1" />
  <attribute name="attr2" value="value2" />
 </attribute-list>
</attribute-matching-queries>
```

All groups having the given attribute value(s) are selected for provisioning.

- <attribute-matching-queries> - Select Groups to be Provisioned by Attribute Resolver

```
<resolver-matching-queries>
  <data-connector-list>
    <data-connector id="ID" />
  </data-connector-list>
</resolver-matching-queries>
```

All groups returned by the GroupDataConnector with the given ID will be provisioned.

- **<groups> - Provision Groups**

```
<groups structure="flat"
        root-dn="ou=grouper,ou=groups,dc=example,dc=edu"
        ldap-object-class="groupOfNames"
        ldap-rdn-attribute="cn"
        grouper-attribute="name">
 <group-members-dn-list ... />
 <group-members-name-list ... />
 <group-attribute-mapping ... />
 <resolver-attribute-mapping ... />
</groups>
```

The optional <groups> element defines how entries and DNs for provisioned groups are created.

<groups>	
structure	The group DN naming structure may be either "flat" or "bushy". A flat structure provisions all groups into the same root DN using the name of the group as the RDN, e.g. cn=stem:child-stem:group-name,root-dn. A bushy structure will provision groups hierarchically, e.g. cn=group-name,ou=child-stem,ou=stem,root-dn.
root-dn	The DN of the entry used as the root of the provisioned groups
ldap-object-class	Defines the LDAP object class used to create each provisioned group. If this object class has required attributes not populated by this provisioning process, then an error will occur.
ldap-rdn-attribute	Defines the attribute in the ldap-object-class used as the RDN. This value may not be "ou" in order to prevent naming collisions between stems and groups when the structure is "bushy".
grouper-attribute	Required when the structure is flat. Defines the attribute value of the group to be used for the value of the ldap-rdn-attribute.
initial-cache-size	Optional attribute specifying the initial size of the group cache. Setting this larger than the likely number of groups to be provisioned should improve performance.
provision-member-groups	If true, member groups should be provisioned as members. Defaults to true. Replaces the "g:gsa" source-subject-identifier.
provision-member-groups-ignore-queries	If true, provision member groups even if they are not in the set of groups to be provisioned. Defaults to false. This is new in v1.5.0, and the behavior of LDAPPC pre-v1.5.0 may be reproduced by setting this to true. In other words, by default, only provision member groups if they are in the set of groups to be provisioned, i.e. match group-queries.

provision-groups-two-step	If true, groups should be provisioned in two steps. The first step provisions all groups without any members. The second step provisions all groups with members. Defaults to true. If false, member groups which have not yet been provisioned may result in log warnings or failures, depending on the value of on-not-found.
bundle-modifications	If true, a group's attribute modifications should be performed in one LDAP operation. If false, each group attribute modification is performed as a separate LDAP operation. Defaults to true.
create-then-modify-members	If true, groups should be created (LDAP add) without members followed by an update (LDAP modify) to add member attributes. Defaults to false.

- <group-members-dn-list> - Provision Member DNs

```
<group-members-dn-list
  list-attribute="member"
  list-object-class="groupOfNames"
  list-empty-value="" />
```

If defined, provisioned groups will include member DNs.

<grouper-members-dn-list	
list-attribute	Defines the LDAP attribute in which to store member DNs.
list-object-class	Optional. Defines the LDAP object class the group entry must have to support the list-attribute. Please note that if this object class has required attributes not populated by this provisioning process, then an error may occur.
list-empty-value	Optional. Defines the value of the list-attribute if no member DNs are stored there. If list-attribute is optional (i.e., a MAY attribute), this value is most likely not needed. If list-attribute is required (i.e., a MUST attribute), then this value should be defined.

- <group-members-name-list> - Provision Member Names

```
<group-members-name-list
  list-attribute="hasMember"
  list-object-class="eduMember"
  list-empty-value="">
 <source-subject-name-mapping>
   <source-subject-name-map source="sourceA" subject-attribute="userid" />
   <source-subject-name-map source="sourceB" subject-attribute="userid" />
</group-members-name-list>
```

If defined, provisioned groups will include member names.

<grouper-members-name-list>	
list-attribute	Defines the LDAP attribute in which to store member names.
list-object-class	Optional. Defines the LDAP object class the group entry must have to support the list-attribute. Please note that if this object class has required attributes not populated by this provisioning process, then an error may occur.
list-empty-value	Optional. Defines the value of the list-attribute if no member DNs are stored there. If list-attribute is optional (i.e., a MAY attribute), this value is most likely not needed. If list-attribute is required (i.e., a MUST attribute), then this value should be defined.

The <source-subject-name-mapping> element contains one or more <source-subject-name-map> elements, which defines the subject attribute containing the subject's name.

<source-subject-name-map>	
source	Source ID
subject-attribute	The Subject attribute containing the Subject's name

- <group-attribute-mapping> - Provision Group Attributes

```
<group-attribute-mapping ldap-object-class="">
 <group-attribute-map
    group-attribute="aci"
    ldap-attribute="aci"
    ldap-attribute-empty-value="" />
</group-attribute-mapping>
```

Optionally, group attributes may be provisioned.

<group-attribute-mapping>	
ldap-object-class	Optional. Defines the LDAP object class the group entry must have to support the attribute mapping. Please note that if this object class has required attributes not populated by this provisioning process, then an error may occur.

The <group-attribute-mapping> element contains one or more <grouper-attribute-map> elements, which map Grouper attributes to LDAP.

<group-attribute-map>	
group-attribute	The Grouper attribute name.
ldap-attribute	The LDAP attribute name.
ldap-attribute-empty-value	Optional. Defines the value to be placed in the ldap-attribute if no values are stored there. If ldap-attribute is optional (i.e., a MAY attribute), this value is most likely not needed. If ldap-attribute is required (i.e., a MUST attribute), then this value should be defined.

- <resolver-attribute-mapping> - Provision Resolver Attributes

```
<resolver-attribute-mapping ldap-object-class="">
 <resolver-attribute-map
    resolver-attribute="sAMAccountName"
    ldap-attribute="sAMAccountName"
    ldap-attribute-empty-value="" />
</resolver-attribute-mapping>
```

Optionally, attributes calculated by the Shibboleth Attribute Resolver may be provisioned. If the <resolver-attribute-mapping> is specified, then three files are required : `ldappc-internal.xml`, `ldappc-services.xml`, and `ldappc-resolver.xml`. These files should be located on the classpath or the directory containing these files may be given as a command line argument. The contents of these files are the same as used by the Shibboleth IDP.

<resolver-attribute-mapping>	
ldap-object-class	Optional. Defines the LDAP object class the group entry must have to support the attribute mapping. Please note that if this object class has required attributes not populated by this provisioning process, then an error may occur.

The <resolver-attribute-mapping> element contains one or more <resolver-attribute-map> elements, which map Shibboleth Attribute Resolver attributes to LDAP.

<resolver-attribute-map>	
resolver-attribute	The Shibboleth Attribute Resolver attribute definition id.
ldap-attribute	The LDAP attribute name.
ldap-attribute-empty-value	Optional. Defines the value to be placed in the ldap-attribute if no values are stored there. If ldap-attribute is optional (i.e., a MAY attribute), this value is most likely not needed. If ldap-attribute is required (i.e., a MUST attribute), then this value should be defined.

- **<membership> - Provision Membership**

```
<memberships>
 <member-groups-list
  list-object-class="eduMember"
  list-attribute="isMemberOf"
  naming-attribute="name"
  temporary-directory="" />
</memberships>
```

In addition to provisioning groups, LDAPPC may provision memberships. The optional <memberships> element contains one <member-groups-list> element, which defines the LDAP attribute of member entries containing the groups of which they are a member.

<member-groups-list	
list-object-class	Optional. Defines the LDAP object class the Member's entry must have to support the group list. Please note that if this object class has required attributes not populated by the provisioning process, then an error may occur.
list-attribute	Defines the LDAP attribute in which to store groups.
naming-attribute	The Grouper attribute used to create the list of groups for a member.
temporary-directory	Optional. Defines the file system directory in which temporary files will be written. Defaults to the current directory.

- **<source-subject-identifiers> - Finding Subjects in the Directory**

```
<source-subject-identifiers>
 <source-subject-identifier
   source="jdbc"
   subject-attribute="id"
   initial-cache-size="350007">
  <ldap-search
    base="ou=people,dc=example,dc=edu"
    scope="onelevel_scope"
    filter="(&(examplePersonId=\{0\})(objectclass=examplePerson)))" />
 </source-subject-identifier>
</source-subject-identifiers>
```

The <source-subject-identifiers> element contains one or more <source-subject-identifier> elements, which defines for a Source the Subject attribute and LDAP search parameters used to lookup Subjects in the directory.

<source-subject-identifier>	**description**
source	Subject Source ID
subject-attribute	The name of the Subject attribute. If a value other than "id" (the subject ID) is specified, performance may suffer as an extra Subject lookup will be performed. It is strongly recommended that the subject ID be in the subject's directory object and that it be indexed.
initial-cache-size	Optional. The initial cache size to cache subject DNs by subject ID. Specifying a larger number than the number of subjects should give better performance.

Each <source-subject-identifier> element contains exactly one <ldap-search> element.

<ldap-search>	**description**
base	The base DN of the context or object to search.
scope	Either "object_scope", "onelevel_scope", or "subtree_scope". The JNDI scope constants are defined in javax.naming.SearchControls. For most flat people branches, "onelevel_scope" is a good choice.
filter	The string "{0}" in the search filter will be replaced by the value of the Subject's attribute defined by subject-attribute in the <source-subject-identifier> element.
on-not-found	Optional, either "warn", "fail", or "ignore". Defaults to "warn". The action that should be taken if the LDAP search does not return any results. "Warn" logs at level WARN. "Fail" throws a RuntimeException which will terminate the LDAPPC process. "Ignore" does nothing.
multiple-results	Optional, either "true" or "false". Defaults to "false". When "false", if multiple results are returned from the LDAP search a RuntimeException is thrown which will terminate the LDAPPC process. When "true", all results returned from the LDAP search will be provisioned.

Example Active Directory Configuration

An example configuration file for provisioning Active Directory might look like the following. There is no `<memberships/>` element since Active Directory handles provisioning the memberOf attribute of group members. In this example, the sAMAccountName attribute, a.k.a. pre-Windows 2000 logon name, is calculated using the Shibboleth Attribute Resolver to replace whitespace in group names with an underscore.

```
<?xml version="1.0" encoding="utf-8"?>

<ldappc>
  <grouper>
    <group-queries>
      <subordinate-stem-queries>
        <stem-list>
          <stem>edu</stem>
        </stem-list>
      </subordinate-stem-queries>
    </group-queries>

    <groups structure="bushy" root-dn="ou=testgroups,${base}" ldap-object-class="group"
            ldap-rdn-attribute="cn" grouper-attribute="name" >

      <group-members-dn-list list-object-class="group" list-attribute="member" />

      <group-attribute-mapping ldap-object-class="group">
              <group-attribute-map group-attribute="description" ldap-attribute="description" />
      </group-attribute-mapping>

      <resolver-attribute-mapping ldap-object-class="group">
        <resolver-attribute-map resolver-attribute="sAMAccountName" ldap-attribute="sAMAccountName" />
      </resolver-attribute-mapping>

    </groups>

  </grouper>

  <source-subject-identifiers>
    <source-subject-identifier source="jdbc" subject-attribute="id">
      <ldap-search base="ou=testpeople,${base}" scope="subtree_scope" filter="(cn={0})" />
    </source-subject-identifier>
  </source-subject-identifiers>

</ldappc>
```

Shibboleth Attribute Resolver configuration :

```
<resolver:AttributeDefinition xsi:type="Script" xmlns="urn:mace:shibboleth:2.0:resolver:ad" id="sAMAccountName"
sourceAttributeID="name">
    <resolver:Dependency ref="groupDataConnector" />
    <Script><![CDATA[
      // Import Shibboleth attribute provider
      value = name.getValues().get(0);

      value = value.replaceAll("\\/", "_");
      value = value.replaceAll("\\/", "_");
      value = value.replaceAll("\\[", "_");
      value = value.replaceAll("\\]", "_");
      value = value.replaceAll("\\:", "_");
      value = value.replaceAll("\\;", "_");
      value = value.replaceAll("\\|", "_");
      value = value.replaceAll("\\=", "_");
      value = value.replaceAll("\\,", "_");
      value = value.replaceAll("\\+", "_");
      value = value.replaceAll("\\*", "_");
      value = value.replaceAll("\\?", "_");

      sAMAccountName = new BasicAttribute("sAMAccountName");
      sAMAccountName.getValues().add(value);
      ]]></Script>
  </resolver:AttributeDefinition>
```

Example OpenLDAP Configuration

An example configuration file for provisioning OpenLDAP might look like :

```
<?xml version="1.0" encoding="utf-8"?>

<ldappc>
  <grouper>
    <group-queries>

      <subordinate-stem-queries>
        <stem-list>
          <stem>_stem_name_</stem>
        </stem-list>
      </subordinate-stem-queries>

      <attribute-matching-queries>
        <attribute-list>
          <attribute name="_attr_name_" value="_attr_value_" />
        </attribute-list>
      </attribute-matching-queries>

    </group-queries>

    <groups
      structure="flat"
      root-dn="ou=groups,${edu.vt.middleware.ldap.base}"
      ldap-object-class="groupOfNames"
      ldap-rdn-attribute="cn"
      grouper-attribute="name">

      <group-members-dn-list list-object-class="groupOfNames" list-attribute="member" list-empty-value="" />

      <group-members-name-list list-object-class="eduMember" list-attribute="hasMember">
        <source-subject-name-mapping>
          <source-subject-name-map source="_source_name_" subject-attribute="_attr_name_" />
          <source-subject-name-map source="g:gsa" subject-attribute="name" />
        </source-subject-name-mapping>
      </group-members-name-list>

      <group-attribute-mapping ldap-object-class="groupOfNames">
        <group-attribute-map group-attribute="description" ldap-attribute="description" />
      </group-attribute-mapping>

    </groups>

    <memberships>
      <member-groups-list list-object-class="eduMember" list-attribute="isMemberOf" naming-attribute="name" />
    </memberships>

  </grouper>

  <source-subject-identifiers>
    <source-subject-identifier source="_source_name_" subject-attribute="_attr_name_">
      <ldap-search
        base="ou=people,${edu.vt.middleware.ldap.base}"
        scope="subtree_scope"
        filter="(uid={0})" />
    </source-subject-identifier>
  </source-subject-identifiers>

</ldappc>
```

Documentation for previous versions is available at https://wiki.internet2.edu/confluence/display/i2miCommon/Ldappc